```
File 348:EUROPEAN PATENTS 1978-2005/Oct W04
         (c) 2005 European Patent Office
File 349:PCT FULLTEXT 1979-2005/UB=20051103,UT=20051027
         (c) 2005 WIPO/Univentio

Set     Items   Description
S1      26932   (16 OR SIXTEEN)(1W)(BIT OR BITS)
S2      14629   (TWO OR 2 OR DUAL)(1W)(BYTE OR BYTES OR CHARACTER? ?)
S3        575   PAIR? ?(3N)(BYTE OR BYTES) OR BYTEPAIR? ?
S4     299077   WORD? ?
S5        888   HALF(2W)WORD? ? OR HALFWORD? ?
S6      44612   (8 OR EIGHT)(1W)(BIT OR BITS)
S7      21141   (ONE OR 1 OR SINGLE OR INDIVIDUAL)(1W)(BYTE OR BYTES OR CH-
             ARACTER? ?)
S8      20139   S1:S4(10N)(FREQUEN? OR STRENGTH? OR OCCURR? OR INCIDENCE? ?
              OR HOW()OFTEN OR APPEAR?)
S9       2539   S5:S7(10N)(FREQUEN? OR STRENGTH? OR OCCURR? OR INCIDENCE? ?
              OR HOW()OFTEN OR APPEAR?)
S10       411   S8(50N)S9
S11        84   S8(50N)S9(50N)(COMPRESS? OR COMPIL? OR DICTIONAR??? OR IND-
             EX???)
S12        33   S11 AND AC=US/PR AND AY=(1970:1999)/PR
S13        33   S11 AND AC=US AND AY=1970:1999
S14        33   S11 AND AC=US AND AY=(1970:1999)/PR
S15        59   S11 AND PY=1970:1999
S16        64   S12:S15
S17        64   IDPAT (sorted in duplicate/non-duplicate order)
```

17/3,K/1   (Item 1 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS

00968474
Method and apparatus for compression and decompression of data
Verfahren und Vorrichtung zur Kompression und Dekompression von Daten
Methode et dispositif pour la compression et la decompression de donnees
PATENT ASSIGNEE:
FUJITSU LIMITED, (211463), 1-1, Kamikodanaka 4-chome, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8588, (JP), (applicant designated states: DE;FR;GB)
INVENTOR:
Yoshida, Shigeru, 3-12-3-205 Morinosato, Atusqi-shi, Kanagawa, 243-0122, (JP)
Nakano, Yasuhiko, c/o Fujitsu Atsugi Ryo, 2-3-10, Sakae-cho, Atsugi-shi, Kanagawa 243, (JP)
Okada, Yoshiyuki, 841-1-612 Minamiyama, Hadano-shi, Kanagawa 257-0003, (JP)
Chiba, Hirotaka, c/o Fujitsu Atsugi Ryo, 2-3-10, Sakae-cho, Atsugi-shi, Kanagawa 243, (JP)
LEGAL REPRESENTATIVE:
Joly, Jean-Jacques et al (39741), Cabinet Beau de Lomenie 158, rue de l'Universite, 75340 Paris Cedex 07, (FR)
PATENT (CC, No, Kind, Date): EP 878915 A2 981118 (Basic)
EP 878915 A3 981230
APPLICATION (CC, No, Date): EP 98201928 910226;
PRIORITY (CC, No, Date): JP 9045163 900226; JP 9062325 900313; JP 9070379 900320; JP 90275835 901015
DESIGNATED STATES: DE; FR; GB
RELATED PARENT NUMBER(S) - PN (AN):
EP 472730 (EP 919043190)
INTERNATIONAL PATENT CLASS: H03M-007/30;
ABSTRACT WORD COUNT: 306

LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	9847	365
SPEC A	(English)	9847	13947
Total word count - document A			14312
Total word count - document B			0
Total word count - documents A + B			14312

...SPECIFICATION for coding in the device of Fig. 31 is illustrated in Fig. 32.

Examples of the unitary **dictionary** and the individual direction when the character sequence "ababcbaba..." are illustrated in Figs. 33 and 34.

An example of the tree of the individual **dictionary** when the character sequence is coded, is illustrated in Fig. 35.

In the example of Fig. 35...

...directly connected to the root of the tree of the last character of the immediately preceding string, **appears** first, the character is taken as raw data and the **single character** is fed.

An example of the code word is shown in Fig. 36. Mode 1 shows the case when the character directly connected to the root of the tree of each individual **dictionary** newly appears.

In mode 1, the combination of the **index** 0, namely designating the raw data, and the raw data of the character is fed as the code word.

When the **index** of the character or string other than 0 **appears** , the **index** in each tree is fed as the code **word** , as shown in Fig. 36.

The flowchart of Fig. 32 will be discussed below.

An initial condition setting step S1 shows the case when 256 individual

dictionaries are provided. However, in order to simplify the disclosure, coding of the character sequence, which the character...

17/3,K/4 (Item 4 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS

00501151
DATA COMPRESSION AND RESTORATION METHOD AND DEVICE THEREFOR
VERFAHREN ZUR KOMPRIMIERUNG UND WIEDERHERSTELLUNG VON DATEN UND GERAT DAZU
PROCEDE DE COMPRESSION ET DE RECONSTITUTION DE DONNEES ET DISPOSITIF PREVU A CET EFFET
PATENT ASSIGNEE:
FUJITSU LIMITED, (211460), 1015, Kamikodanaka, Nakahara-ku, Kawasaki-shi, Kanagawa 211, (JP), (Proprietor designated states: all)
INVENTOR:
YOSHIDA, Shigeru, 218-3, Kawaraguchi, Ebina-shi, Kanagawa 243-04, (JP)
NAKANO, Yasuhiko, Fujitsu Atsugi Ryo, 2-3-10, Sakae-cho, Atsugi-shi, Kanagawa 243, (JP)
OKADA, Yoshiyuki, City Haitsu Ohkid 405, 256-1, Tanaka, Isehara-shi, Kanagawa 259-11, (JP)
CHIBA, Hirotaka, Fujitsu Atsugi Ryo, 2-3-10, Sakae-cho, Atsugi-shi, Kanagawa 243, (JP)
LEGAL REPRESENTATIVE:
Joly, Jean-Jacques et al (39741), Cabinet Beau de Lomenie 158, rue de l'Universite, 75340 Paris Cedex 07, (FR)
PATENT (CC, No, Kind, Date): EP 472730 A1 920304 (Basic)
EP 472730 A1 921216
EP 472730 B1 000510
WO 9113395 910905
APPLICATION (CC, No, Date): EP 91904319 910226; WO 91JP252 910226
PRIORITY (CC, No, Date): JP 9045163 900226; JP 9062325 900313; JP 9070379 900320; JP 90275835 901015
DESIGNATED STATES: DE; FR; GB
RELATED DIVISIONAL NUMBER(S) - PN (AN):
EP 871294 (EP 98201925)
EP 871295 (EP 98201926)
EP 878915 (EP 98201928)
INTERNATIONAL PATENT CLASS: G06F-005/00; H03M-007/30
ABSTRACT WORD COUNT: 192

LANGUAGE (Publication,Procedural,Application): English; English; Japanese
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS B	(English)	200019	560
CLAIMS B	(German)	200019	524
CLAIMS B	(French)	200019	670
SPEC B	(English)	200019	14379
Total word count - document A			0
Total word count - document B			16133
Total word count - documents A + B			16133

...SPECIFICATION for coding in the device of Fig. 31 is illustrated in Fig. 32.
Examples of the unitary **dictionary** and the individual direction when the character sequence "ababcbaba..." are illustrated in Figs. 33 and 34.

An example of the tree of the individual **dictionary** when the character sequence is coded, is illustrated in Fig. 35.
In the example of Fig. 35...

...directly connected to the root of the tree of the last character of the immediately preceding string, **appears** first, the character is taken as raw data and the **single** **character** is fed.

14/9/7 (Item 7 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)

01102655 SUPPLIER NUMBER: 03542259 (THIS IS THE FULL TEXT)
Data compression **techniques.**
McCarthy, Michael
Computers & Electronics, v22, p67
Dec, 1984
ISSN: 0745-1458 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT
WORD COUNT: 3378 LINE COUNT: 00254

TEXT:

How would you like to nearly double the effective ASCII file storage capacity of your archive media, without hardware modifications, and without having to pay for the required software?

OR, if you send ASCII files between machines by modern, how would you like to cut transmission times almost in half? Again, for free!

Are you interested?

Of course you are. At least for CP/M and MS-DOS, programs that are currently available in the public domain will let you accomplish these things. For other operating systems you can purchase such programs as Documax for the Apple.

If you're using CP/M or MS-DOS and want the benefits of this software but don't care about the underlying technology, all you need to do is ask your neighborhood CP/M or MS-DOS expert how you can get copies of:

SQ.COM (which **compresses** an ASCII file to about 60% of its original size) and USQ.COM (which recreates the original file from its **compressed** version).

Then ask about:

LU.COM (which gathers several files into a single library file to save more disk space),

LRUN.COM (which fetches a .COM file from a library and runs it),

LTYPE.COM (which types a [**compressed**] library text file).

But read on if you want to understand something about the way these programs work, how they were developed, and what are some novel applications of the ideas that underlie them.

What is Data **Compression** ?

Data **compression** is a way of sorting information so that it takes up much less space than it would using standard storage techniques. Because the data takes up less space, transmitting it from one place to another can also save time.

The benefit is more data capacity per unit of physical storage (or more data transmitted per unit time). However, there being no such thing as a free lunch, there are technical prices to pay.

First, because the storage techniques are nonstandard, hardly any applications or utilities are able to cope with **compressed** data. so you need utilities to convert data, not only from standard to **compressed** form, but also back again. In addition, running these utilities takes time.

Second, **compressed** data is less reliable than standard data: Imagine that you have an ordinary ASCII file. If a single bit in the file goes bad, then an occurence of the letter "c", say, might be changed to the letter "b." With the effect of a typo, the error often can be harmless (assuming that your operating system will continue to let you access the file).

The corresponding error in a **compressed** version of the same file would be a disaster. As you will see later, the damage would alter not only the "garbaged" letter but also every letter after it, in such a way that only an expert could reclaim it. But first let's explore what data **compression** is.

F U Cn Rd Ths

Cn y ndrstnd ths prgrph F crs y can Thts bcs y dnt nd vwls r spcl chrctrs t ndrstnd wrttn Nglsh

(Can you understand this paragraph? Of course you can. that's because you don't need vowels or special characters to understand written English.)

Notice that by eliminating the vowels we have reduced the text to about 70% of its original size. The italic paragraph is an example of a (crude) data **compression** scheme.

It works because most of the information in written English (or any written European language) is carried by the consonants. The vowels and punctuation add very little to the information content. They are in most cases nearly "redundant" and dispensable. All data **compression** schemes eliminate some kind of redundancy.

the **compression** procedure we used above was very simple:

1. Get the next character.
2. If the character is a space, keep it.
3. If the character belongs to the set a,E,I,O,U, discard it.
4. If the character is nonalphabetic, discard it.
5. If the character is anything else, keep it.

This " **compression** " procedure is easily programmed. However, no one today knows how to program a computer to perform the corresponding "decompression." We humans simply don't know how to express what the rules are, even though we use them intuitively.

For example, take the "nd" in the phrase "dnt nd vwls." Must the "nd" represent "need?" Could it be "and?" Could it have been an abbreviation for "industrial?" How about "nod" or "node" or "nude?"

you know that the word must have been "need" because only this reconstruction fits the context--but no program today could have made that judgment because no program truly understands English (even though some very smart people have been working on this kind of problem for more than 30 years).

The result is that for a computer environment we are going to need a different method, one that can be programmed for decompression as well as for **compression** . The **compression** procedure will have to identify redundancy and eliminate it. The procedure for reconstruction will have to be able to regenerate the original text precisely. Neither procedure will be permitted to rely on more than superficial knowledge of natural languages. We need to find a technique that **compresses** while it retains the full original text, character-for-character.

As you will see, we will end uo with an outline design for a utility program that accomplishes our goal while offering a **compression** of 40-50%. Best of all, the utility that does it exists and is free, at least for CP/M and MS-DOS systems. (It's the SQ, for "SQueezer," utility mentioned earlier.)

The Development of the Concepts

Data **compression** is as old as the telegraph. Telegraphs that transmitted text characters electrically may first have been used in Germany quite early in the 19th century. However, a telegraph that included a complete system for the economical transmission of text messages was invented (and more important, then developed) by Samuel F. B. Morse. Morse built a whole business around the goal of message transmission. He decided to base his charges on message word counts. Then, he designed a procedure to minimize the time required to send a message. (He assumed, correctly, that the service would be very popular and would be heavily loaded, with transmission line capacity being the limiting factor.) His procedure, of course, was "Morse code."

A clever feature of Morse's code is that it exploits the letter-frequency characteristics of the English language. By and large, the most commonly used letters in English have the shortest encodements. for example, the letter "e" is used more often than any other, and it is the shortest symbol in Morse, consisting of a single "dot." In contrast, the Morse encodement of the infrequently used "q" is "dash-dash-dot-dash," which takes 13 times longer to send than the dot for "e."

Now, Morse was very clever, but he wasn't entirely scientific or mathematical in his approach. There is a betteer way, and he missed it. (But then, so did everybody else until around 1950.) This better way involves using a Huffman code, which generalizes and formalizes the concept of letter frequencies. It is particularly well-suited to computers.

What Data **Compression**

Programs Do

The public domain program SQ.COM incorporates a Huffman code for shortening the number of bits required to represent text. (It also gives special treatment to repeating sequences of characters." Overall, SQ typically is able to "squeeze out" about 40% or so of an original ASCII file.

Its decompression counterpart, USQ.COM ("USQ" for "UnSQueezer") takes filed produced by SQ.COM and restores them to their original ASCII format, bit-for-bit.

Both SQ and USQ work quite well. They are easy to use; they are reliable; and their documentation is more than adequate. Their author, Dick Greenlaw, did an excellent job.

If you're tight on storage space and need SQ/USQ, then chances are excellent that you are also short of directory space and are limited in the number of files you can create (unmodified CP/M 2.2 is limited to 64 directory entries). Accordingly, you also want the LU (Library Utility) program along with its suite of related programs.

LU and its relatives allow you to package a collection of separates files into a single "library" file with its own embedded (sub)directory. The files may be **compressed** or not. The related programs allow you to load and execute a .COM file embedded in a library, to type a **compressed** library subfile, and so on. LU also has the side benefit of conserving disk space; short files waste much less space when in library files than when they are stored on an individual basis. All in all, this software is valuable, well worth tracking down.

How Data **Compression** Works

Suppose that we are dealing with a stream of characters represented by 8-bit bytes (which is how ASCII text is actually stored in your computer). In principle, any single byte could contain one of 256 possible values, shown here in 8-bit binary:

In practice, however, ASCII files consist of text and/or numbers and/or special characters, like "$." Even counting letters twice (for upper and lower case), we get along very nicely with a total of only 96 symbols to be represented.

Indeed, a rudimentary **compression** scheme would simply express the 96 codes actually used not with 8 bits, but with only 7 bits (which allows for 128 possibilities).

But this method saves us only 1 bit in 8, or 12.5%. We can do better than this--much better--if we assume that we are dealing with text, because we can exploit the idea of letter frequencies.

We'll invoke Pareto's law, named after the Renaissance Italian mathematician who formulated a rule about relative frequencies of things in human events. He noticed that 80% of the revenue of a typical business tended to be generated by 20% of its customers, 80% of its inventory value was tied up in 20% of the inventory items, and so on.

Well-known as the "80/20 rule," this idea has inestimable value to businesspeople and managers. The philosophical importance of the 80/20 rule is that if you do a good job on 20% of the issues, you will be solving 80% of your problems. (Whence the phrase "80% solution.")

Applying Pareto's law to text **compression** , if we do a good job of **compressing** the 20% of characters that make up 80% of our text, then the overall scheme very likely will be reasonably efficient, even if no attempt is made to optimize for the relatively infrequent characters that make up the remaining 80% of the possible characters.

To a first approximation, the four most commonly encountered symbols will be, in decreasing order of frequency:

Ignoring for the moment how to encode all the other characters, we will encode just these four, in binary, as follows:

This is a variable-length code. The first bit tells us whether the symbol being represented is "space." If it isn't "space" (first bit isn't 1), then the second bit tells us whether it's "e." If it isn't "e" (second bit isn't 1), then the third bit tells us whether it's "t" ... and so on through all the characters of interest, in order of decreasing frequency.

This code works neatly. If we assume that for English text the average word length is five characters, then every sixth character or so will be a space. Thus about 1/6, or 17%, of our characters will be

represented by a single bit instead of 8 bits.

The letter "e" will not be as common as "space," but we still will require only 2 bits to express it (rather than 8), so we're still way ahead.

Now it is easy to see that we could extend this idea indefinitely, taking 5 bits for the 5th most common character, 6 bits for the 6th most common, and so on.

However, it also seems intuitively clear that, at least for English text, it would be dumb to use 9 bits to represent the 9th most common character, given that we can represent any character at all with just 8.

Evidently, at some point we need to change gears and deviate from a strict Huffman code. In fact, let's make the changeover point 4 bits, just as outlined above. Now the encodement for anything except "space," "e," "t," or "a" will begin with 0000.

Now for a crucial trick:

After 0000, let's always use all of the next 4 bits to represent a single character. In those next 4 bits we can express 16 possible values. Let's use 15 of those 16 values to represent the 5th through 19th most common characters.

What about the 16th possible 4-bit value? We'll use it as an "escape code." Let's say that the escape code is 0000. Then whenever we see 0000 (not "space/e/t/a") followed by another 0000, we have a signal that whatever is to be represented is not one of the 19 most common characters.

How small we represent lower-frequency characters? By Pareto's law we might as well represent such infrequent characters with their real full 8-bit codes. Such situations occur sufficiently infrequently that they are not worth optimizing.

To summarize: We use anywhere from 1 to 4 bits to represent each of the four most common characters. For any of these characters the win is big, a saving of anywhere from 4 to 7 bits (when compared to a full 8-bit ASCII code).

We use 8 bits to represent the 5th through 19th most common ones. Here there is no win, but no loss either.

Finally, we use 16 bits (0000 followed by 0000 followed by a full 8-bit code) to represent everything else. Using these, of course, is tremendously inefficient when we consider such characters by themselves. But when we look at the overall statistics, we win big (1-4 bits) much more often than we lose big (16 bits).

Now let's add one more level of complexity for dealing with repetitious sequences of a given character. Instead of using one escape code, let's use two. The first escape code has the meaning earlier described. The additional escape code is for repetition. It means: Take the next 4 bits as a count; take the 4 bits after that as a Huffman code (with the option to expand to 8 or 16 bits as necessary).

This works well for lengthy repetition sequences of the same character, typically spaces. (For short sequences it would be better to encode the spaces as a short sequence of 1's.)

The overall scheme will win only if the four most common characters drastically dominate the actual letter frequencies. Fortunately they do. In fact, the frequencies are so biased in favor of the four most common characters (or repeated spaces) that the resulting "bit stream" tends to be around half the length of the original stream of 8-bit ASCII characters.

Adaptive Encoder Program

The length will be halved for most English text, but it may not be true for other languages or for other applications (such as storing program source files). We can get around this limitation by building an "adaptive encoder" program that will work with any ASCII file.

The adaptive encoder would make two passes over the ASCII file to be **compressed** . During the first pass it would read the file, building a table of the 256 possible ASCII byte values and counting the number of times each value occurred in the original ASCII file.

The encoder then would select the four most common values and assign them to the four variable-length codes. The next 14 characters that **appeared** most **frequently** would be assigned to the set of **8 - bit** codes. Finally, all the remaining characters would be assigned to the set

of **16 - bit** codes.

Armed with the actual **frequency** of **occurrence** of each character, during the second pass the encoder would create its output file, making the substitutions decided upon at the end of the first pass.

Note that the file doesn't have to be English text. It could be any language. For that matter, it could even be business data consisting mostly of spaces and numbers. The adaptive encoder will simply observe the relative frequencies of the various characters and decide on the most efficient encoding scheme.

All that matters is that the distribution of character occurrences be highly "skewed," with a few characters drastically outweighing all of the others in frequency of occurrence.

The decoding (decompression) process is equally straightforward. The only complication is that the decoder must agree with the encoder as to the meaning of each of the symbol types (variable-length, 8-bit, 16-bit). This agreement is reached by having the encoder record its substitution table at the front of the encoder's output file. The decoder reads this table and decides what the decoding rules are before reading (and expanding) the **compressed** information itself.

Other Issues

The encoder's output file will usually be significantly shorter than the original input file. I say "usually" because there are two circumstances under which the output file could actually end up being longer than the input file.

If the four most common byte values are not drastically more common than all the others, then the output file will be longer than the input file. The "something else" values will be too frequent (an almost certain occurrence if the file being **compressed** is an executable binary program file. For highly technical reasons such files are nearly pseudo-random, with few outstandingly frequent unique byte values).

Also, the encoder's substitution table takes up a certain amount of space in the output file. If the input file were very short, the combination of frequency table and encoded data could be longer than the input data, even if that input data were otherwise **compressible** ASCII text.

The **compressed** file is extremely sensitive to error because with variable-length Huffman coding there no longer is a one-to-one relationship between bytes in the original data and bytes in the **compressed** file. The corruption of even a single data bit can wreck the file.

For example, consider the word "at," encoded as described above:

0001001XXXXXXXXXXXXXXXXXXXX

a t remainder of text

If the fourth bit is "dropped" and thereby changed to zero, the decoder will not have "seen" the original "a." In fact, it will overrun the "t" because of the rule that says "if you see 0000, then take the next 4 bits as ..."

The decoder is now hopelessly desynchronized with respect to the original input. It will, in fact, expand the remainder of the bit stream to gibberish.

For reasons of this nature, military and commercial communication and storage systems that employ data **compression** schemes invariably incorporate an error-correcting scheme to ensure integrity of the **compressed** data.

Interesting Reading

An excellent treatment of both Huffman codes and error correction is given in Richard Hamming's Coding and Information theory (Prentice-Hall, 1980).

You should be aware that this marvelous book requires a good grounding in algebra and probability theory. The mathematics is unavoidable, but if you can cope with it, the book provides a valuable overview of information theory and its important major subtopics. Unlike most writers in the field of error correction, Hamming aspires to educate readers rather than to "snow" them. He succeeds, but you have to work with him--there is no "royal road."

15/3,K/21 (Item 21 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)

01062621 SUPPLIER NUMBER: 00593472
Sanyo MBC 1150.
Edelsen, R.H.
Interface Age, v9, n6, p87-88
June, 1984
DOCUMENT TYPE: evaluaton ISSN: 0147-2992 LANGUAGE: ENGLISH
RECORD TYPE: ABSTRACT

...ABSTRACT: and monitor. Documentation is thorough, with only a few small problems. The MBC 1150 is an effective **8 - bit** microcomputer. However, as more and more **16 - bit** machines **appear** , it may become outdated. A photograph of the MBC 1150 is included.

15/3,K/22 (Item 22 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)

01042281 SUPPLIER NUMBER: 02948721 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Baud-rate matching for CoCo.
Harding, S.E.
Computers & Electronics, v21, p85(2)
Oct, 1983
ISSN: 0745-1458 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT
WORD COUNT: 652 LINE COUNT: 00046

... is fed to IC2A pin 20.

When the buffer within the receiver is full (it contains an **8 - bit** byte), a load-OK signal **appears** at pin 22 of IC2A to inform the computer that it is time to send another character. This is called "handshaking."

The parallel data **word** from the receiver buffer **appears** on pins 5 through 12 which are directly coupled to the parallel input buffer of the transmitter...

15/3,K/23 (Item 23 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)

01022453 SUPPLIER NUMBER: 00502520
CompuPro System 816-C.
Fox, T.
Interface Age, v8, n7, p61-63
July, 1983
DOCUMENT TYPE: evaluaton ISSN: 0147-2992 LANGUAGE: ENGLISH
RECORD TYPE: ABSTRACT

ABSTRACT: CompuPro's System 816-C computer is designed to take advantage of **16 - bit** software that is just starting to **appear** in the marketplace, and also run existing thoroughly tested **8 - bit** software in two ways: by including an 8085 8-bit microprocessor and by using Intel's 8088...

15/3,K/24 (Item 1 from file: 621)
DIALOG(R)File 621:Gale Group New Prod.Annou.(R)

01817249 Supplier Number: 53972204 (USE FORMAT 7 FOR FULLTEXT)
Mitsubishi Expands M16C Line with Two New 16-bit Families, Product Line Bridges Gap Between 8-bit and 32-bit Applications.

Business Wire, p1046
March 1, 1999
Language: English   Record Type: Fulltext
Document Type: Newswire; Trade
Word Count: 1131

...   be quickly processed.
The M16C/80 has also improved on its predecessor's C language code efficiency. **Frequently** used **16 - bit** instructions are allocated to a **one - byte** instruction, reducing code length and execution cycles. The M16C/80 includes a 24-bit addressing mode, a...

15/3,K/25   (Item 2 from file: 621)
DIALOG(R)File 621:Gale Group New Prod.Annou.(R)

01602996   Supplier Number: 48264814   (USE FORMAT 7 FOR FULLTEXT)
Samsung Debuts New Line of Smart Card ICs.
Business Wire, p02020088
Feb 2, 1998
Language: English   Record Type: Fulltext
Document Type: Newswire; Trade
Word Count: 678

...   data buffer memory to facilitate authentication and serial communications with the data terminal.
Other features include a **16 - bit** random number generator and voltage and **frequency** detect modules for increased security, as well as an **8 - bit** basic timer for on-chip housekeeping and terminal data synchronization.
Samsung's chip operating system (COS) conforms...

15/3,K/26   (Item 3 from file: 621)
DIALOG(R)File 621:Gale Group New Prod.Annou.(R)

01426950   Supplier Number: 46710524   (USE FORMAT 7 FOR FULLTEXT)
Siemens Announces Controller Area Network Version of its C515 8-BIT Microcontroller.
Business Wire, p09160160
Sept 16, 1996
Language: English   Record Type: Fulltext
Document Type: Newswire; Trade
Word Count: 352

...   6 microsecond conversion time integrated on the chip.
A 600ns instruction execution time at 10MHz together with **eight 16 - bit** data pointers ensures maximum performance even during **frequent** external data fetches. An interrupt unit with 15 possible sources on four priority levels allows an improved...

15/3,K/27   (Item 4 from file: 621)
DIALOG(R)File 621:Gale Group New Prod.Annou.(R)

01230841   Supplier Number: 44097090   (USE FORMAT 7 FOR FULLTEXT)
SPT Announces 8-Bit, 300 & 150 MSPS Flash A/D Converters
News Release, p1
Sept 14, 1993
Language: English   Record Type: Fulltext
Document Type: Magazine/Journal; Trade
Word Count: 921

...   dynamic performance and sampling rate specifications. The SPT7725
    digitizes a two-volt analog signal with full-scale **frequency**
    components to 210 MHz into **8 - bit** digital **words**
at conversion rates of
    up to 300 megasamples per second (MSPS). Signal-to-noise ratio (SNR)
    with...

15/3,K/28 (Item 5 from file: 621)
DIALOG(R)File 621:Gale Group New Prod.Annou.(R)

01207956 Supplier Number: 43505867 (USE FORMAT 7 FOR FULLTEXT)
8/16-Bit Embedded Controllers Have High Speeds: 3V Version Runs at 10 MHz; 5V Unit, at 16 MHz
News Release, p1
Dec 7, 1992
Language: English Record Type: Fulltext
Document Type: Magazine/Journal; Trade
Word Count: 829

...   H8/536s from
    Hitachi America Ltd., Semiconductor & I.C. Division. Whether
    operating from 3V or 5V, these **8 / 16 - bit** embedded controllers now
    operate with substantially higher clock **frequencies** .

    With a 3V supply, the maximum clock speed increases to 10 MHz from 5
    MHz; with a...

15/3,K/29 (Item 6 from file: 621)
DIALOG(R)File 621:Gale Group New Prod.Annou.(R)

01131932 Supplier Number: 41111456 (USE FORMAT 7 FOR FULLTEXT)
ADC-32/33, 8-BIT 150 MHZ, FLASH A/D CONVERTER
News Release, p1
Jan 11, 1990
Language: English Record Type: Fulltext
Document Type: Magazine/Journal; Trade
Word Count: 274

...   150 MHZ, FLASH A/D CONVERTER

    DATEL's ADC-32 and ADC-33 are monolithic, video speed, **8 - bit** flash
    converters capable of digitizing analog signals with **frequency**
    spectral components to 100MHz, into digital **words** at a 150 MSPS
    conversion rate.

    External sample-and-hold is not required for proper operation of...

15/3,K/30 (Item 7 from file: 621)
DIALOG(R)File 621:Gale Group New Prod.Annou.(R)

01120565 Supplier Number: 40930552 (USE FORMAT 7 FOR FULLTEXT)
PC-BASED CRYSTAL CONTROLLED ARBITRARY WAVEFORM GENERATOR
News Release, p1
Sept 5, 1989
Language: English Record Type: Fulltext
Document Type: Magazine/Journal; Trade
Word Count: 551

... programmable Digital data are available on a 9
pin header on the Printed Circuit Board, thereby providing **16 bit** output. These additional **8 bits** can have a maximum clock **frequency** of
40 MHz, giving a Digital frequency of 20 MHz and a maximum pattern depth of 16...

15/3,K/31 (Item 8 from file: 621)
DIALOG(R)File 621:Gale Group New Prod.Annou.(R)

01053099 Supplier Number: 40175540 (USE FORMAT 7 FOR FULLTEXT)
GE INTEGRATES RCA AND INTERSIL DSP AND DATA CONVERSION PRODUCTS AT CUPERTINO FACILITY
News Release, pN/A
Sept 29, 1987
Language: English Record Type: Fulltext
Document Type: Magazine/Journal; Trade
Word Count: 431

... standard 16-bit multiplier/accumulator, is
focused on building block and applications-oriented solutions in the medium- **frequency** /high-precision market in the 1 mHz/ **16 - bit** range.
GE's products targeted the high- **frequency** /lower-precision video applications in the 10-20 mHz/ **8 - bit** range.

"These DSP products will allow us to penetrate certain high-speed market segments sooner than we...

15/3,K/32 (Item 9 from file: 621)
DIALOG(R)File 621:Gale Group New Prod.Annou.(R)

01050470 Supplier Number: 40146553 (USE FORMAT 7 FOR FULLTEXT)
IDT's NEW FLASH A/D CONVERTER PERFORMS AT TWICE THE SPEED AND USES HALF THE POWER
PR Newswire, pN/A
August 26, 1987
Language: English Record Type: Fulltext
Document Type: Newswire; Trade
Word Count: 654

... rate of 20 MagaSamples per
Second (MSPS). It is capable of converting analog signals with full power **frequency** components up to 7 MHz into **8 - bit** digital **words**
.
Input sample and hold circuits are not required. Unlike bipolar ADC devices, the IDT device has an...

15/3,K/33 (Item 1 from file: 636)
DIALOG(R)File 636:Gale Group Newsletter DB(TM)

03813175 Supplier Number: 48269991 (USE FORMAT 7 FOR FULLTEXT)
HEWLETT-PACKARD: Electromechanical design & process characterization costs lowered
M2 Presswire, pN/A
Feb 3, 1998
Language: English Record Type: Fulltext
Document Type: Newswire; Trade
Word Count: 559

... reluctance and digital state. The HP E1419A has multiple control capabilities, including analog voltage and current outputs, **8 - bit** or **16 - bit** digital outputs, pulse output with variable **frequency** , pulse-width modulation, watchdog timeout, crank angle and stepper motor control.

HP VEE is a visual environment...

15/3,K/34 (Item 2 from file: 636)
DIALOG(R)File 636:Gale Group Newsletter DB(TM)

02532788 Supplier Number: 45105136 (USE FORMAT 7 FOR FULLTEXT)
Siemens Siemens is one of the world's leading suppliers of electrical and electronic products, systems and equipment.
Integrated Circuits International, pN/A
Nov, 1994
Language: English Record Type: Fulltext
Document Type: Newsletter; Trade
Word Count: 1485

(USE FORMAT 7 FOR FULLTEXT)
TEXT:
...which have already been completed. Major growth areas for Siemens are expected to be communications ICs, high **frequency** GaAs chips, derivatives of DRAM technology, chip card ICs, **8 / 16 - bit** microcontrollers and very high power semiconductors. Quarterly financial results for Siemens show the company on schedule to...

15/3,K/35 (Item 3 from file: 636)
DIALOG(R)File 636:Gale Group Newsletter DB(TM)

02418198 Supplier Number: 44803395 (USE FORMAT 7 FOR FULLTEXT)
SIEMENS' SEMICONDUCTOR GROUP TO GROW 25%
Integrated Circuits International, pN/A
July, 1994
Language: English Record Type: Fulltext
Document Type: Newsletter; Trade
Word Count: 113

(USE FORMAT 7 FOR FULLTEXT)
TEXT:
...next year (fiscal 1994/5). Major growth areas for Siemens are expected to be communications ICs, high **frequency** GaAs chips, derivatives of DRAM technology, chip card ICs, **8 / 16 - bit** microcontrollers and very high power semiconductors. Contact: Klaus H. Knapp, Siemens Semiconductors, tel/fax: (49) (89) 4144...

15/3,K/36 (Item 4 from file: 636)
DIALOG(R)File 636:Gale Group Newsletter DB(TM)

02410816 Supplier Number: 44780614 (USE FORMAT 7 FOR FULLTEXT)
ADVANCED LOGIC HAS DIGITAL WAVE KIT TO ADD MULTIMEDIA TO ANY 80386 OR HIGHER
Computergram International, n2442, pN/A
June 23, 1994
Language: English Record Type: Fulltext
Document Type: Newswire; Trade
Word Count: 275

(USE FORMAT 7 FOR FULLTEXT)
TEXT:
...the company says gives it the audio output capabilities of a professional MIDI synthesiser. The board's **frequency** range is 20Hz to 22KHz, but it also offers **8 - bit** and **16 - bit** digital sampling at **frequencies** up to 48KHz. Advanced Logic has also launched a dual-speed CD-ROM drive which it says...

15/3,K/37 (Item 5 from file: 636)
DIALOG(R)File 636:Gale Group Newsletter DB(TM)

02064890 Supplier Number: 43782756 (USE FORMAT 7 FOR FULLTEXT)
New ASSP Unit Targets MCUs
Semiconductor Industry & Business Survey, v15, n5, pN/A
April 19, 1993
Language: English Record Type: Fulltext
Document Type: Newsletter; Trade
Word Count: 214

(USE FORMAT 7 FOR FULLTEXT)
TEXT:
...of Microchip's current marketing strategy can be summed up in three letters: MCU. Or in two **words** : embedded control. To **strengthen** its market position in **8 - bit** field programmable MCUs and allied specialty memories, the company has formed a new Application Specific Standard Product...

15/3,K/38 (Item 6 from file: 636)
DIALOG(R)File 636:Gale Group Newsletter DB(TM)

01129470 Supplier Number: 40880500 (USE FORMAT 7 FOR FULLTEXT)
VERBEX VOICE TRADER FOR DEAL CAPTURE
Trading Systems Technology, v3, n3, pN/A
July 31, 1989
Language: English Record Type: Fulltext
Document Type: Newsletter; Trade
Word Count: 664

... into the Voice Trader handset and say "Buy 100 shares of General Electric at 56 and a **half** -- done." The **words** are then processed by the system and **appear** on the screen of a PC linked to the Voice Trader output. The system is always open...

15/3,K/39 (Item 7 from file: 636)
DIALOG(R)File 636:Gale Group Newsletter DB(TM)

01007612 Supplier Number: 40304897 (USE FORMAT 7 FOR FULLTEXT)
FUJITSU DEVELOPS HIGH-SPEED 4-BIT SLICE MICROPROCESSOR+
Japan Semiconductor Scan, pN/A
Feb 24, 1988
Language: English Record Type: Fulltext
Document Type: Newsletter; Trade
Word Count: 270

... BIT SLICE MICROPROCESSOR, SO CALLED BECAUSE IT IS DESIGNED TO BE LINKED IN A SERIES TO PRODUCE **8 - BIT** OR **16 - BIT** MICROPROCESSORS, HAS A CLOCK **FREQUENCY** OF 770 MEGAHERTZ (A MEGAHERTZ EQUALS 1 MILLION CYCLES PER SECOND), WHICH IS 25 TIMES FASTER THAN...

15/3,K/40 (Item 1 from file: 16)
DIALOG(R)File 16:Gale Group PROMT(R)

05463952 Supplier Number: 48281718 (USE FORMAT 7 FOR FULLTEXT)
Samsung intros smart-card ICs
Electronic Buyers' News, p025
Feb 9, 1998
Language: English Record Type: Fulltext
Document Type: Magazine/Journal; Trade
Word Count: 331

... terminal. The device features a CPU core based on a 16-bit address bus, and includes a **16 - bit** random- number generator and voltage and **frequency** detect modules for increased security, as well as an **8 - bit** basic timer for on-chip housekeeping and terminal data synchronization, according to Samsung.

The chip's operating...

15/3,K/41 (Item 2 from file: 16)
DIALOG(R)File 16:Gale Group PROMT(R)

03032069 Supplier Number: 44120174 (USE FORMAT 7 FOR FULLTEXT)
SPT INTRODUCES MONO FLASH A-D CONVERTER
Electronic News (1991), p24
Sept 27, 1993
Language: English Record Type: Fulltext
Document Type: Magazine/Journal; Trade
Word Count: 246

The SPT7725 digitizes a 2-volt analog signal with full-scale **frequency** components to 210MHz into **8 - bit** digital **words** at conversion rates up to 300 megasamples per second (MSPS). Signal-to-noise ratio (SNR) with a...

15/3,K/42 (Item 3 from file: 16)
DIALOG(R)File 16:Gale Group PROMT(R)

02832184 Supplier Number: 43808286
Japanese, Korean Makers Enter DRAM Accord
Office Equipment & Products, p28
May, 1993
Language: English Record Type: Abstract
Document Type: Magazine/Journal; Trade

ABSTRACT:
...DRAM. The 2 firms will initially make 4 versions of 16-Mbit synchronous DRAMs consisting of 2M- **words** x **8** **bits** and operating at **frequencies** of 66- and 100-MHz. ...

15/3,K/43 (Item 4 from file: 16)
DIALOG(R)File 16:Gale Group PROMT(R)

02422870 Supplier Number: 43190164 (USE FORMAT 7 FOR FULLTEXT)
PORTABLE POWER
Chain Store Age Executive with Shopping Center Age, p72
August, 1992
Language: English Record Type: Fulltext

Document Type: Magazine/Journal; Trade
Word Count: 112

(USE FORMAT 7 FOR FULLTEXT)
TEXT:
...Development, Natick, Mass., reveals that 52.3% of U.S. shipments of batch-mode PDCTs now feature **8 - bit** or **16 - bit** microprocessors. Rapidly-evolving radio **frequency** (RF) technology has also enhanced the utility of the devices by allowing users to operate in real...

15/3,K/44 (Item 5 from file: 16)
DIALOG(R)File 16:Gale Group PROMT(R)

02081926 Supplier Number: 42693252 (USE FORMAT 7 FOR FULLTEXT)
Prices Fall, Demand Rises For 16-Bit Adapter Cards
Systems & Network Integration, p22
Jan 27, 1992
Language: English Record Type: Fulltext
Document Type: Magazine/Journal; Trade
Word Count: 971

... at what the competition is doing."
16-Bit Growth
Standard Microsystems has found that while demand for **8 - bit** Ethernet adapter cards remains strong, the major growth is **occurring** in its **16 - bit** line. According to Kristine Stewart, director of channel marketing at Standard Microsystems, there is a lot of...

15/3,K/45 (Item 6 from file: 16)
DIALOG(R)File 16:Gale Group PROMT(R)

01750824 Supplier Number: 42193887 (USE FORMAT 7 FOR FULLTEXT)
Tradewest Takes Road to Hollywood
HFD-The Weekly Home Furnishings Newspaper, v0, n0, p80
July 1, 1991
Language: English Record Type: Fulltext Abstract
Document Type: Magazine/Journal; Trade
Word Count: 547

... emulating the feel of an arcade," Cook said. "We try to duplicate it within the limits of **8 - bit** and **16 - bit** systéms."
The exception to the strategy **appears** to be Battletoads. The game grew from a agreement with Rare Inc. and has spawned a series...

15/3,K/46 (Item 7 from file: 16)
DIALOG(R)File 16:Gale Group PROMT(R)

01259451 Supplier Number: 41463501 (USE FORMAT 7 FOR FULLTEXT)
NEC sees late '91 as watershed for 16-bit: Company projects majority control for newest generation of game devices
HFD-The Weekly Home Furnishings Newspaper, v0, n0, p87
July 30, 1990
Language: English Record Type: Fulltext Abstract
Document Type: Magazine/Journal; Trade
Word Count: 867

ABSTRACT:
...bit versions marketed by Sega and Nintendo of America Inc.
Even Nintendo, which has staunchly backed its **8 - bit** system, **appears**

poised to enter the **16 - bit** arena. In June, Nintendo Co., the parent company of the American subsidiary, unveiled its 16-bit Super...
... bit versions marketed by Sega and Nintendo of America Inc.
Even Nintendo, which has staunchly backed its **8 - bit** system, **appears** poised to enter the **16 - bit** arena. In June, Nintendo Co., the parent company of the American subsidiary, unveiled its 16-bit Super...

15/3,K/47 (Item 1 from file: 160)
DIALOG(R)File 160:Gale Group PROMT(R)

01639688
Analog Devices Mkts Converter.
ELECTRONIC NEWS May 11, 1987 p. 46

... 5-V TTL or 12-V CMOS. Data are available within a maximum of 45-ns in **2 bytes** over an **8 - bit** bus. The device's reference **frequency** ranges from 400-Hz to 20,000-Hz. ...

15/3,K/48 (Item 2 from file: 160)
DIALOG(R)File 160:Gale Group PROMT(R)

01346007
NEC Markets High-Speed Line Memories.
JOURNAL OF THE ELECTRONICS INDUSTRY March, 1986 p. 74

... uPD41101C operates at a read write/write cyle time of 34-ns or 8X the chroma subcarrier **frequency** in an optimum cell arrangement of 10 **words** x **8 bits** .

15/3,K/49 (Item 3 from file: 160)
DIALOG(R)File 160:Gale Group PROMT(R)

00622885
Microprocessor makers say they are handling the transition from 16- to 32-bit systems better than they handled the switch from 8- to 16-bit machines.
Computer Business News January 5, 1981 p. 1,5

... software compatibility between their 16-bit machines and the future 32-bit generations--a compatibility that is **frequently** missing between **8 - bit** machines and **16 - bit** products. Microprocessor makers won't introduce the 32-bit machines until the market is really ready to...

15/3,K/50 (Item 1 from file: 148)
DIALOG(R)File 148:Gale Group Trade & Industry DB

10117511 SUPPLIER NUMBER: 20457761 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Microcomputer measures high-frequency signals.
D'Souza, Stan
EDN, v43, n5, p106(1)
March 2, 1998
ISSN: 0012-7515 LANGUAGE: English RECORD TYPE: Fulltext; Abstract
WORD COUNT: 704 LINE COUNT: 00056

... prescaler value. You can modify Figure 1a's circuit using a PICmicro (micro) C to implement a **16 - bit** **frequency** counter (Figure 1b).

(Figure 1a-1b ILLUSTRATION OMITTED)

This circuit uses the internal **8 - bit** prescaler to divide the incoming **frequency** . The circuit feeds the output of the prescaler to the **8 - bit** timer, (TMR.sub.0), for measurement. As with Figure 1a's circuit, once the gate time is...

...until the 8-bit timer/counter, (TMR.sub.0), increments by 1. In this case, the lower **8 - bit** value of the measured **frequency** equals 256-N. The (micro) C then concatenates the value of the counter with the 8-bit timer's value to give a **16 - bit** value of the measured **frequency** .

Figure 1c shows a further simplification of the circuit in Figure 1b by replacing the NAND gates...

```
File 275:Gale Group Computer DB(TM) 1983-2005/Nov 08
         (c) 2005 The Gale Group
File 621:Gale Group New Prod.Annou.(R) 1985-2005/Nov 09
         (c) 2005 The Gale Group
File 636:Gale Group Newsletter DB(TM) 1987-2005/Nov 09
         (c) 2005 The Gale Group
File  16:Gale Group PROMT(R) 1990-2005/Nov 09
         (c) 2005 The Gale Group
File 160:Gale Group PROMT(R) 1972-1989
         (c) 1999 The Gale Group
File 148:Gale Group Trade & Industry DB 1976-2005/Nov 09
         (c)2005 The Gale Group
File 624:McGraw-Hill Publications 1985-2005/Nov 08
         (c) 2005 McGraw-Hill Co. Inc
File  15:ABI/Inform(R) 1971-2005/Nov 09
         (c) 2005 ProQuest Info&Learning
File 647:CMP  Computer Fulltext 1988-2005/Oct W5
         (c) 2005 CMP Media, LLC
File 674:Computer News Fulltext 1989-2005/Oct W2
         (c) 2005 IDG Communications
File 696:DIALOG Telecom. Newsletters 1995-2005/Nov 08
         (c) 2005 Dialog
File 369:New Scientist 1994-2005/Jul W3
         (c) 2005 Reed Business Information Ltd.

Set      Items    Description
S1      116904    (16 OR SIXTEEN)(1W)(BIT OR BITS)
S2        7151    (TWO OR 2)(1W)(BYTE OR BYTES)
S3         232    PAIR? ?(3N)(BYTE OR BYTES) OR BYTEPAIR? ?
S4     1585712    WORD? ?
S5       67235    (8 OR EIGHT)(1W)(BIT OR BITS)
S6       16075    (ONE OR 1 OR SINGLE OR INDIVIDUAL)(1W)(BYTE OR BYTES)
S7       37032    S1:S4(10N)(FREQUEN? OR STRENGTH? OR OCCURR? OR INCIDENCE? ?
                OR HOW()OFTEN OR APPEAR?)
S8        1080    S5:S6(10N)(FREQUEN? OR STRENGTH? OR OCCURR? OR INCIDENCE? ?
                OR HOW()OFTEN OR APPEAR?)
S9         731    HALF(2W)WORD? ? OR HALFWORD? ?
S10         15    S9(10N)(FREQUEN? OR STRENGTH? OR OCCURR? OR INCIDENCE? ? OR
                HOW()OFTEN OR APPEAR?)
S11        168    S7(50N)(S8 OR S10)
S12        116    RD (unique items)
S13        104    S12 NOT PY=2000:2005
S14         22    S13 AND COMPRESS?
S15         82    S13 NOT S14
S16       7150    (TWO OR 2 OR DUAL)(1W)CHARACTER? ?
S17        513    PAIR? ?(3N)CHARACTER? ?
S18      10388    (ONE OR 1 OR SINGLE OR INDIVIDUAL)(1W)CHARACTER? ?
S19        230    S16:S17(10N)(FREQUEN? OR STRENGTH? OR OCCURR? OR INCIDENCE?
                ? OR HOW()OFTEN OR APPEAR?)
S20        222    S18(10N)(FREQUEN? OR STRENGTH? OR OCCURR? OR INCIDENCE? ? -
               OR HOW()OFTEN OR APPEAR?)
S21        176    (S7 OR S19)(50N)(S8 OR S10 OR S20)
S22        124    RD (unique items)
S23          8    S22 NOT S12
```